



Building The European Network for Lifelong Competence Development

Building the European Network
For Lifelong Competence Development

TENCompetence IST-2005-027087

Project Deliverable Report

Internal deliverable nr 8.19 Social support portlet

Workpackage	8
Task	Taskforce 17
Date of delivery	Contactual: 01-10-2009 Actual: 22-12-2009
Code name	Version: 0.9 Draft <input type="checkbox"/> Final <input checked="" type="checkbox"/>
Type of deliverable	Report
Security (distribution level)	Internal
Contributors	SU, OUNL
Authors (Partner)	Francis Brouns (OUNL), Adelina Aleksieva-Petrova, Milen Petrov (Sofia University)
Contact Person	Francis Brouns
WP/Task responsible	INSEAD
EC Project Officer	
Abstract (for dissemination)	Social help Liferay portlet implementation
Keywords List	

TENCompetence Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 45 5762624 – Fax: +31 45 5762800

1. Introduction.....	3
2. Scenario.....	3
3. Portlet implementation.....	5
4. Portlet API	7
4.1 Activities/Portlets.....	7
4.2. Flow of events (design phase).....	8
4.2.1. Flow of events for do Request Use Case	8
4.2.2. Flow of events for Refuse/Agree to Participate In Discussion Use Case	8
4.2.3. Flow of events for Participate In Discussion Use Case	9
4.3. Objects	9
4.3.1. SocialHelpUser	10
4.3.2. SocialHelpRequest.....	11
4.3.3. SocialHelpWorkflow	11
4.3.4. SocialHelpForum	11
4.4. Invitation Scheduler Service	12

Social support portlet

1. Introduction

The social support portlet provides a learner support service in which other learners in the network are engaged in providing assistance to learners who have a particular request for support. After a learner has formulated a question, the TENCompetence infrastructure assists in finding the most suitable person(s) to answer this particular question. The social support portlet caters for the following kind of scenarios.

2. Scenario

Suppose we have a Community on Psychology with a competence profile for Educational psychology. User Philip has registered for this community, selected the CP Educational psychology and created a personal development plan (PDP). Finally, let's assume that Philip while studying A1 Quantitative data analysis, runs into problems. He has a problem understanding the relations between a number of concepts and as a consequence he is not able to complete an assignment. He studies some additional literature and searches the web, to no avail though. Philip is studying on his own and thus out of touch with any peer students decides to pose a question to the 'on-line tutor'; he describes the general problem and his question. The TENCompetence system then selects those peers that would be the best candidate for assisting Philip and sets up an ad hoc transient community to allow Philip and his peers to arrive at an answer to his question.

Actors: Learners and peers, system

Flow of Events

1. Philip has registered to the community and created a basic profile, containing at least his name and email address.
2. Philip has selected the competence profile, matching competence development plan, performed a self-assessment and created a personal development plan.
3. While working for action A1, Philip has difficulty understanding some concepts. The resources in the action do not provide sufficient detail or are of the wrong level to help Philip in finding the answer himself. He decides to look for support.
4. Philip accesses the support form that is available from the action and poses his question in sufficient detail.
5. The system selects the most suitable peers and invites them to assist Philip in finding an answer to his question.
6. The peers can accept or decline this invitation.
7. When the required number of peers has accepted the invitation, the system sets up a forum that can be accessed by Philip and the selected peers. The forum contains the question, and a guideline.
8. The system notifies Philip and the peers about the existence of the forum.

9. Philip and peers can discuss the question, and jointly reach a solution or answer to the question.

A generalized workflow is depicted in Figure 1.

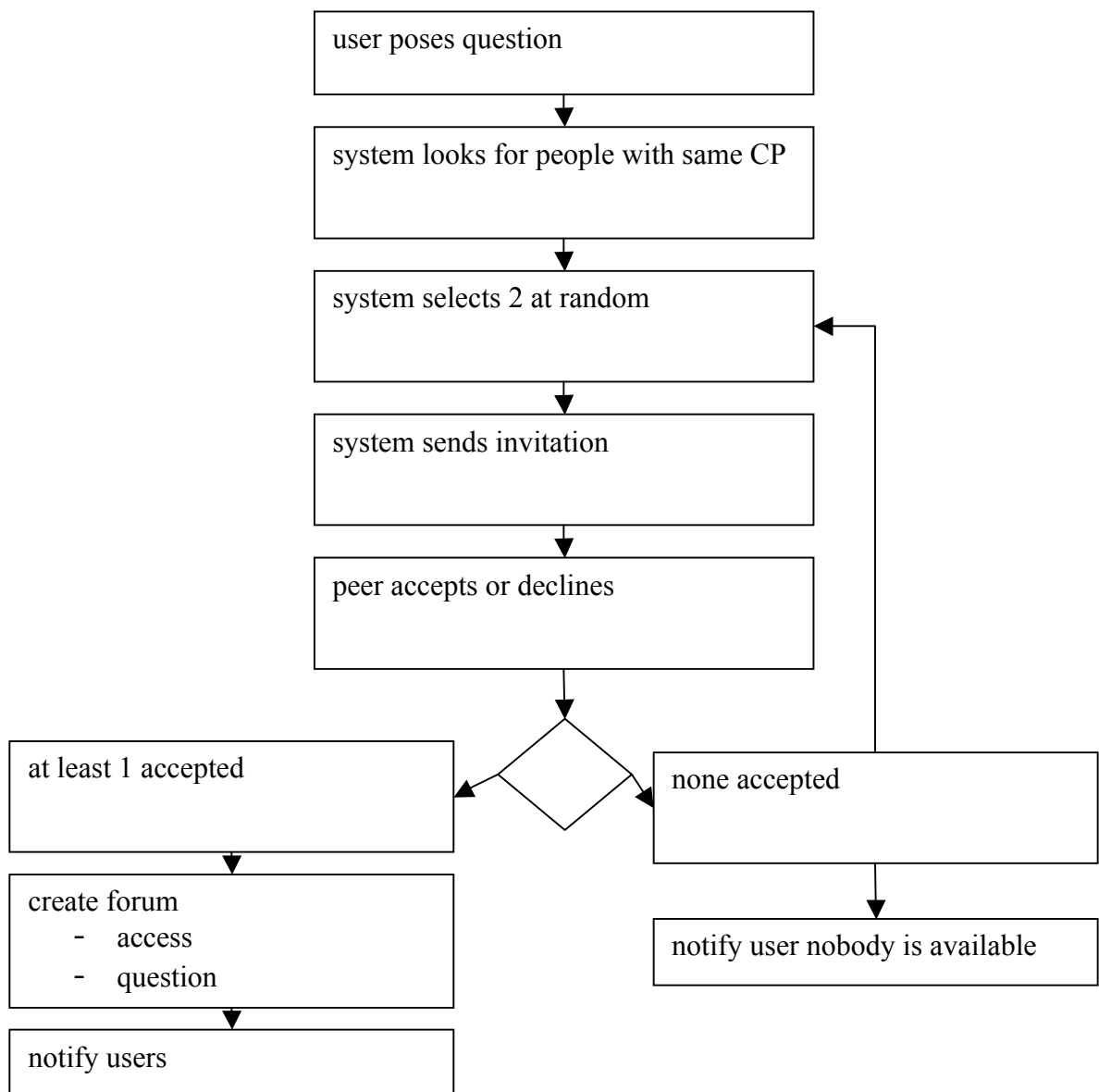


Figure 1: Flow of events for social support

3. Portlet implementation

A first release of Liferay social support portlet has been implemented. The source code is available under the BSD licence, copyright TENCompetence Foundation, from Sourceforge (<http://sf.net/projects/tencompetence>). The first release is also available from the DSpace repository at <http://hdl.handle.net/1820/2121>

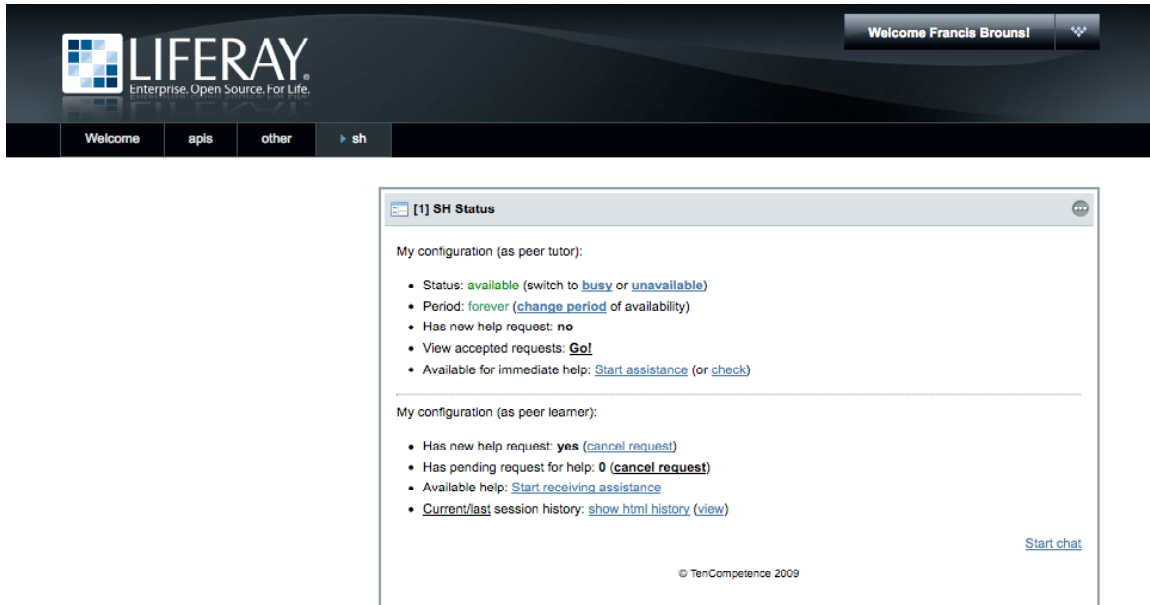
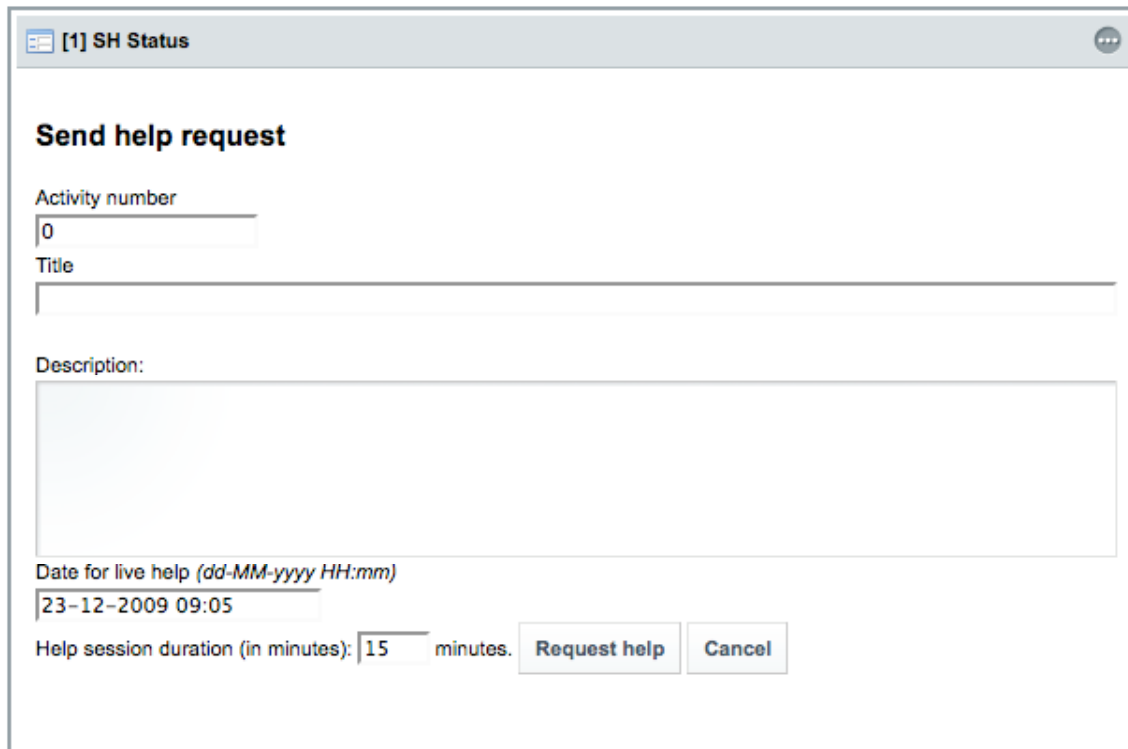


Figure 2: Overview of social help portlet options



The screenshot shows a web portlet titled "[1] SH Status". Below the title is a section labeled "Send help request". It contains several input fields: "Activity number" with the value "0", "Title" (empty), and "Description:" (empty). Below these is a "Date for live help (dd-MM-yyyy HH:mm)" field with the value "23-12-2009 09:05". At the bottom, there is a "Help session duration (in minutes):" field with the value "15" and the text "minutes.". To the right of this field are two buttons: "Request help" and "Cancel".

Figure 3: Form to formulate the request

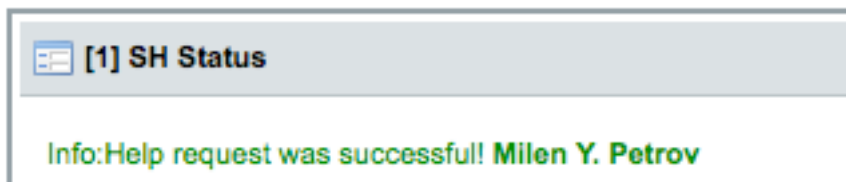
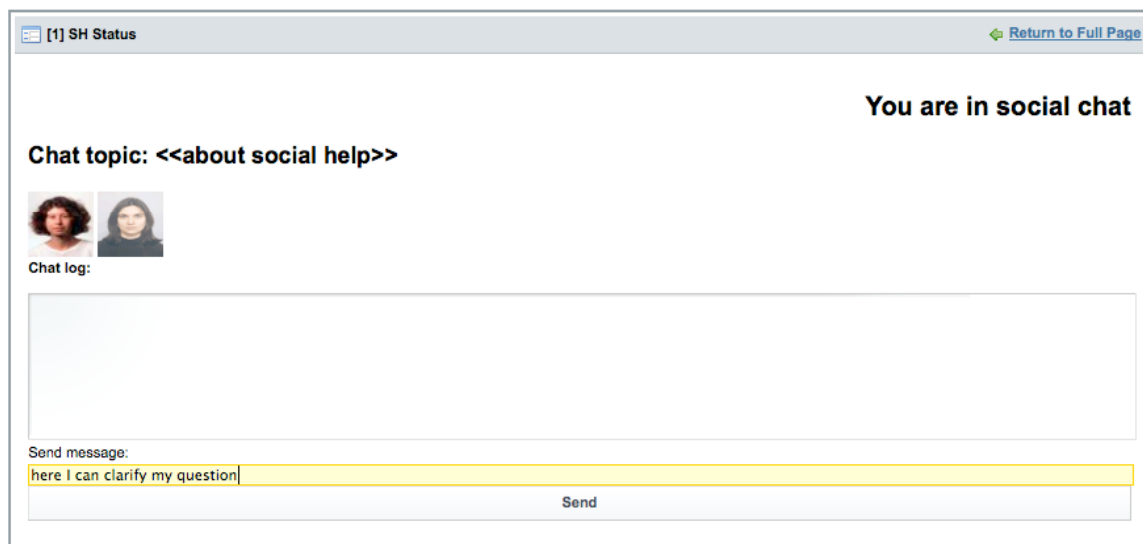


Figure 4: Request was successful



The screenshot shows a web portlet titled "[1] SH Status" with a "Return to Full Page" link in the top right corner. The main content area is titled "You are in social chat". Below this is a "Chat topic: <<about social help>>" section. There are two profile pictures of users. Below the pictures is a "Chat log:" section with a large empty text area. At the bottom, there is a "Send message:" section with a text input field containing the text "here I can clarify my question" and a "Send" button.

Figure 5: Discuss with peer to obtain answer

4. Portlet API

The technical design is described in the form of the API.

4.1 Activities/Portlets

This section contains a list of activities that LNUs (learning network users) may perform. These activities are split up into three separate portlets:

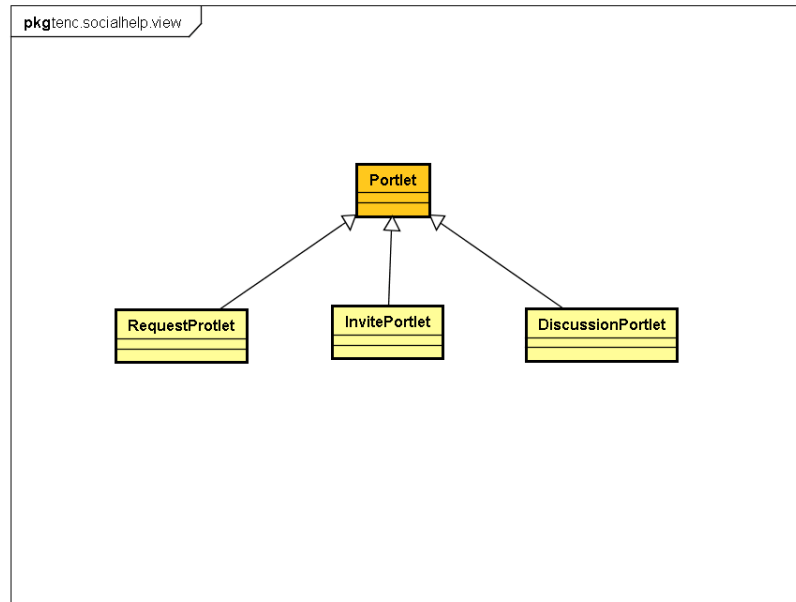


Diagram 1: Portlet architecture

1. **Request Portlet** – This portlet serves requests to the system to start the social help procedure. *Request* is sent by the *LNUs* and is not visible from *LNUs*.

This portlet launch the follow activities:

- Define *specific Request* to some problem.
 - Set all *LNUs* as potential *peer tutor* participating in a *Social Help*
 - Launch the jobSearchSchedule Portlet
2. **Invite Portlet** – this portlet executes the search algorithm for tutor suitability. It creates a ranked list of users and selects the first two of them. Then executes the job which invites the selected peer tutors by mail. This job and task has an exactly specified time (2 days). These are persistent jobs, for which the state is saved in a database and it can be sure that those jobs won't be lost. The invitation cycle has reached completion when some peer tutor accepts the invitation.

The tutor gets an invitation by e-mail. The message contains a description of the problem and corresponding activity. The tutor may either refuse or agree to join. If a tutor were to either accept or reject after expiration of the invitation, he or she should receive a message to the effect that the invitation has expired.

3. **Discussion Portlet** - The *Discussion Portlet* is used for holding *Discussions*. Every *Request* initiates a new *Thread* of the *Discussion*. The *Thread* consists of a multitude of *Messages* containing information about the *Request* of the respective *Thread*. *Discussions* may be added by the *LNU* with make the *Request*. *Threads* are formed when a *LNU* sends a *Message*. The *LNU* may also reply to an existing *Message*. This way he continues the *Thread*. Then it (*The Thread*) becomes a hierarchy of *Messages* – sent and replied. *Messages* have a title and contain a short text. They must also keep information about their sender and the sending date.

4.2. Flow of events (design phase)

4.2.1. Flow of events for do Request Use Case

Precondition:

The user has logged to the system and is recognized as a system *LNU*.

Main flow:

1. The UI creates a Request Processor instance.
2. The Request Processor instance is initialised.
3. The UI provides the content of the *Request*.
4. A new instance of the Request Content database object is created.
5. The Request Processor saves into the database and launches the search algorithm and job schedule for sending invitation mail to peer tutors.

4.2.2. Flow of events for Refuse/Agree to Participate In Discussion Use Case

Precondition:

The user has logged to the system and is recognized as a system *LNU*.

Main flow:

1. The *LNU* follows a link to the *Views Requests* from his/her control page.
2. The system shows the main page of the *Active Requests*. It contains all current *Requests* for this *LNU*.
3. The *LNU* selects a *request* and view problem's details.
4. The *LNU* may Refuse or Agree the request invitation.

4.2.3. Flow of events for Participate In Discussion Use Case

Precondition:

The user has logged to the system and is recognized as a system *LNU*.

Main flow:

LNU may participate in any *Problem Discussion* that exists in the *Social Help Discussion Board*.

1. The *LNU* follows a link to the *Social Help Discussion Board* from his control page.
2. The system shows the main page of the *Social Help Discussion Board*. It contains all current *Problem Discussion*.
3. The *LNU* selects a *request* and launches it.
4. The system shows all requests (*Threads*) available.
5. The *LNU* may send (S1) either a new *Message*, or browse the tree of *Messages* and reply to any of them. After sending the *Message*, the system goes back to (4).

Subflows:

S1. The system shows a new page – a *Message* composer. There the *LNU* types the text of the message. Finally the *LNU* sends the *Message*.

4.3. Objects

In this section the objects needed for the Social Help are described. It should be noted that the lists of object fields listed below may not be exhaustive and it include only specific fields. Social Help API should provide getters and/or setter for all fields listed in this section.

The main Social Help API objects are showed in follow class diagram:

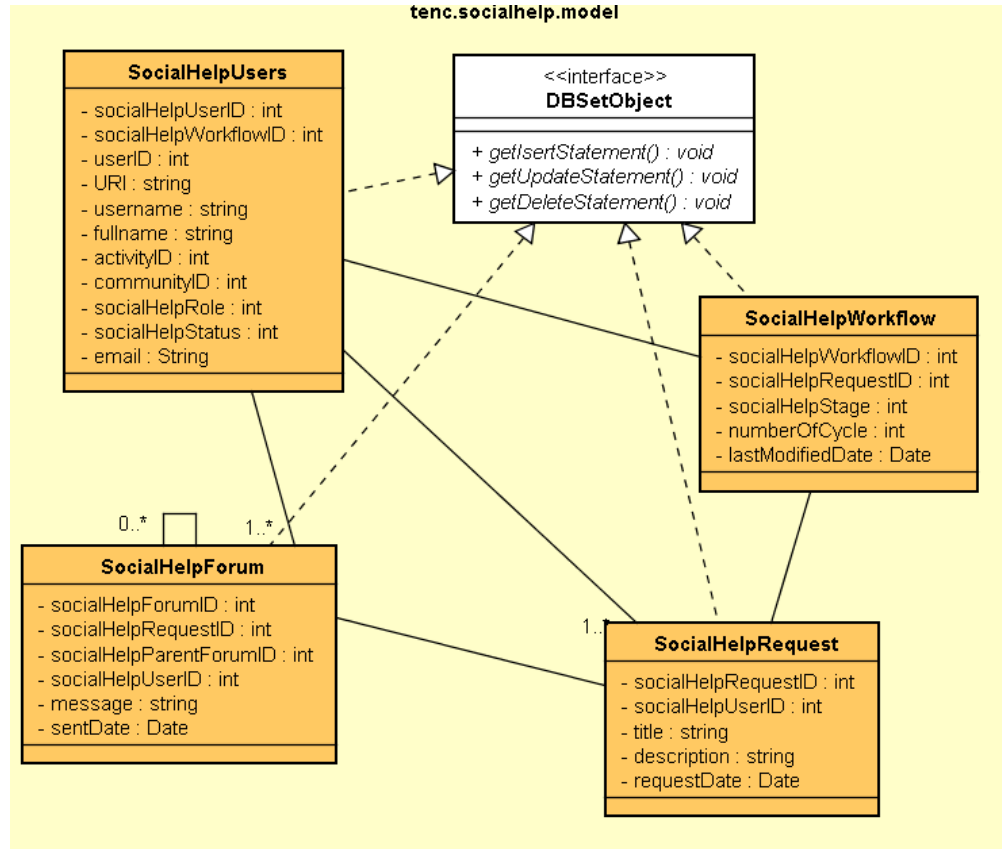


Diagram 2: Class diagram social help user

4.3.1. SocialHelpUser

This section contains an extension of the definition of the *user* object in invitation cycle.

Field Name	Values	Default Value	M/A ¹	Reason/Meaning/Usage
Email	String	""	M	E-mail address of the user
socialHelpRole	Integer	0	A	Indicates what is the user role: (1) learner and (2) peer tutor.
socialHelpStatus	Integer	0	A	The status has follow value: (1) – receive invitation; (2) – accept invitation; (3) – decline invitation.

¹ This column describe whether the associated field is either filled in 'Manually, by the user, or 'Automatically, by the system.

The list of methods of this class follows:

Method Name	Static	Return	Parameters		Notes
			Name	Type	
receiveInvitation	N	Boolean	question	Integer	<p>This method should be used to invite a user to participate as a tutor in a peer community.</p> <p>If the user accepts the invitation, the method returns true and false, otherwise.</p>

4.3.2. SocialHelpRequest

This object represents the user request for social help.

Field Name	Values	Default Value	M/A	Reason/Meaning/Usage
socialHelpRequestID	Integer	Last ID + 1	A	A unique identifier for each agenda
socialHelpUserID	Integer	UserID	A	Uniquely identifies, the LNU, associated with the Request.
discription	String	""	M	The description of the problem.
title	String	""	M	The title of problem request.

4.3.3. SocialHelpWorkflow

This object represent the invitation cycle.

Field Name	Values	Default Value	M/A	Reason/Meaning/Usage
SocialHelpWorkflow ID	Integer	Last ID + 1	A	A unique identifier for the tutor competence object.
socialHelpRequestID	Integer	0	A	Provides identifier from Request
numberOfCycle	Integer	0	A	The number of invitation cycle.

4.3.4. SocialHelpForum

This object is used to represent a problem discussion forum.

Field Name	Values	Default Value	M/A	Motivation/Reason/Usage
socialHelpForumID	Integer	Last ID + 1	A	Provides a unique identifier for forum

socialHelpRequestID	Integer	0	A	Provides identifier from Request
socialHelpUserID	Integer	0	A	Provides identifier from user
socialHelpParentForumID	Integer	0	A	Provides identifier from parent forum. It is hierarchical structure of data.
sendDate	Long	Now()	A	Indicates the date and time of sent the message.
message	String	""	M	Defines the subject of the message.

4.4. Invitation Scheduler Service

The Invitation Scheduler service would ensure that jobs (send invitation email) are scheduled to run at specific times in the future. These jobs could be run multiple times based on the user's preference.

The Invitation Scheduler Service that we are going to develop will have the following features:

- The ability to schedule jobs at fixed and varying times
- The ability to schedule jobs that can run at fixed intervals indefinitely
- The ability to cancel jobs
- The ability to list all the currently scheduled jobs

The figure below shows the sequence of events of the Invitation Scheduler Service.

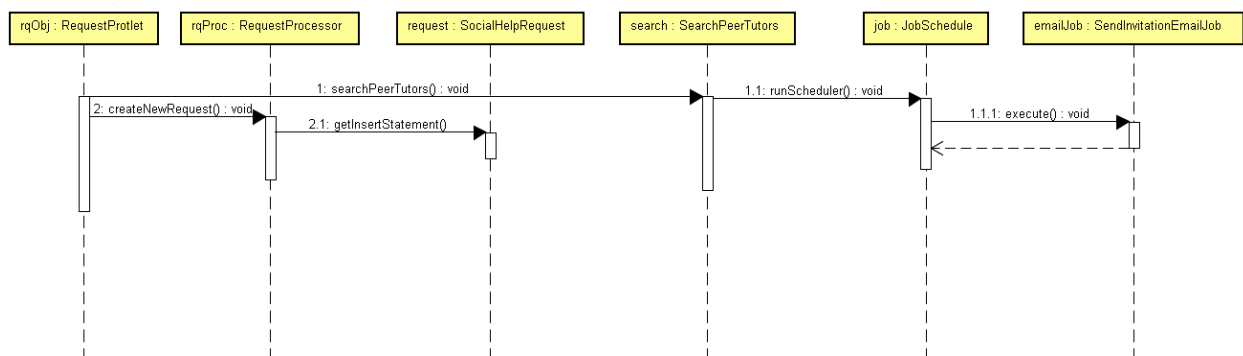


Diagram 3: Invitation Scheduler Service